MAR-61-E-040695
68000 Sound Driver Ver. 3.00
SEGA Enterprises, Ltd.

68000 Sound Driver
Ver. 3.00

Introduction

The 68000 Sound Driver Version 3.03 (hereinafter "V3") is designed to fully exploit the sound source that is incorporated into the SUPER32X.

Sound Driver Specifications

The following are the specifications for the Sound Driver:

Target sound sources	FM5 or 6-sound/PSG tone 3+ noise 1/PWM 4-sound PCM0-2 sound (8-bit linear or SEGA 4-bit delta format)
Required resources Driver size (68000) Work size CPU load (68000) IDOL Music-playing SE (per channel) CPU load (Z80)	 Approximately 3000H (including a Z80 driver (*1)) Approximately 0B00H Approximately 1% Approximately 9% Approximately 0.5% 100% (exclusive use)
Number of tracks Music SE	 14-16 tracks (varies with the PCM driver) 5 tracks
Remarks	A software envelope is provided as a PSG. Two types of software vibrato functions are provided as an FM/PSG. These resources endow all sound sources with the same "look and feel". A drum kit can be created as an FM sound source.

(*1) If PCM is not used, the driver size decreases by approximately 1300H.

Activating the Sound Driver

The following procedures are used to control the V3:

1) Use a system call (more on this later) to initialize the Sound work space.
2) Call the starting address for the Sound Driver every V-INT (approximately 16 ms).

See the system call column later in this chapter for a method for requesting sound data.

Memory Map

The Sound Driver operates under the following memory map:

ROM memory map
Sound Driver [approximately 3000H]
Sound Driver system unit
Data vectors (offset address) [4x8 bytes]
PCM data [undefined]
PWM data [undefined]
Music data [undefined]
SE data [undefined]
Table envelope data (undefined, in units of 100H bytes)
Table vibrato data (undefined, in units of 100H bytes)
Rhythm kit data (undefined, in units of 8 bytes)
FM sound source data (undefined, in units of 25 bytes)

RAM Memory Map

Common work space [60-70H (varies with the number of PCM sounds generated)]
Channel work space [60H x channels (for all music and SE)]
YM-2612 write buffer [200H bytes]

Fill-In Data

The Driver is filled with the following information at a distance of 0CH from the starting address:

Meaning	Size
Work space starting address	4 bytes
Work space size	4 bytes
PCM type (00H: none, 10H: linear, 20H: delta)	1 byte
Number of sounds generated from the PCM sound source	1 byte
Number of sounds generated from the FM sound source	1 byte
Number of sounds generated from the PSG sound source	1 byte
Number of sounds generated from the PWM sound source	1 byte
Total number of tracks	1 byte
Total number of tracks per music	1 byte
Total number of tracks per SE	1 byte
Driver version number	6 bytes
Driver type	Undefined

System Calls

The V3 uses system calls to support driver controls. This enables the V3 to accommodate driver version upgrades, as well as work-space changes, without requiring a modification of the V3 itself. Although it is possible to control the driver by directly rewriting the work space, as has been done in the past, to provide for future version upgrades it is recommended that the driver be controlled by means of system calls.

List of System Calls

Number	Operation	Input register	Output register	Destroyed register
00H	Initialize the Sound work space and the hardware around the Sound Driver.	D1	None	None
01H	Requests a piece of music.	D1	None	D0/A0
02H	Requests an SE.	D1	None	D0/A0
03H	Sets a fade-in/out.	D1	None	None
04H	Sets the music master volume.	D1	None	None
05H	Sets the SE master volume.	D1	None	None
06H	Sets the music master transpose.	D1	None	None
07H	Sets the SE master transpose.	D1	None	None
08H	Sets a pause.	None	None	None
09H	Resets a pause.	None	None	None
0AH	Writes communication data.	D1	None	None
0BH	Reads communication data.	None	D1	None
0CH	Requests to stop the music.	None	None	D0/A0

0DH	Requests to stop the SE.	None	None	None

Details on System Calls

First, set the desired system call number in the register D0. Then, set register values as necessary, and call the address that is obtained by adding 8 to the starting address of the Sound Driver.

00H Initializing the Sound Driver and the hardware around the Sound Driver

Function: Initialize the Sound Driver and the hardware around the Sound Driver.

Input: D0.b ... 00H
D1.b ... sets NTSC/PAL
0 ... sets to NTSC
1 ... sets to PAL

Output:

Destruction:

Remarks:

01H Requesting a piece of music

Function: Performs music.

Input: D0.b ... 00H
D1.b ... piece number

Output:

Destruction: D0/A0

Remarks: Because of a 4-byte request buffer, sounds can be produced in a maximum of four simultaneous interrupts (common to the SE number). Some request numbers can cause the SE to be performed (music numbers are not checked).

02H Requesting an SE

Function: Produces SE sounds.

Input: D0.b ... 02H
D1.b ... SE number

Output:

Destruction: D0/A0

Remarks: Because of a 4-byte request buffer, sounds can be produced in a maximum of four simultaneous interrupts (common to the music piece number). Some request numbers can cause music to be performed (music numbers are not checked).

03H Fade-request

Function: Sets the fade-in/out option.

Input: D0.b ... 03H
D1.w ... sets the fade value.

15~8 bit	7~0 bit
Fading depth	Fading speed

Output:
Destruction:
Remarks: Fade-in operators can be performed by setting the fading depth to a negative number (2's complement). The allowable range of fading speed is 00H-7FH (in units of V-int).

04H Setting the master music volume
Function: Sets the volume for the entire music.
Input: D0.b ... 04H
D1.b ... volume
Output:
Remarks: The allowable range of volume is 00H-7FH, where 00H indicates the maximum volume, and 7FH the mute option.

05H Setting the master SE volume
Function: Sets the volume for the entire SE.
Input: D0.b ... 05H
D1.b ... volume
Output:
Destruction:
Remarks: The allowable range of volume is 00H-7FH, where 00H indicates the maximum volume, and 7FH the mute option.

06H Setting the master music transpose
Function: Transposes the entire music.
Input: D0.b ... 06H
D1.b ... transposition value
Output:
Destruction:
Remarks: A transposition value can be a negative value (2's complement).

07H Setting the master SE transpose
Function: Transposes the entire SE.
Input: D0.b ... 07H
D1.b ... transposition value
Output:
Destruction:

Remarks: A transposition value can be a negative value (2's complement).

08H Requesting a pause

Function: Sets a pause.
Input: D0.b ... 08H
Output:
Destruction:
Remarks: Because PWM sound sources lack a function for stopping sound production on a channel-by-channel basis, once sound production is started it cannot be stopped.

09H Resetting a pause

Function: Resets a pause.
Input: D0.b ... 09H
Output:
Destruction:
Remarks:

0AH Writing communication data

Function: Writes communication data into sound work space.
Input: D0.b ... 0AH
D1.b ... data
Output:
Destruction:
Remarks: This is a processing action viewed from the main system. Currently data cannot be read on the Driver. Therefore, writing data from the main system will not alter the processing.

0BH Reading communication data

Function: Reads communication data from sound work space.
Input: D0.b ... 0BH
Output: D0.b ... data
Destruction:
Remarks: This is a processing action viewed from the main system.

0CH Requesting to stop music

Function: Terminates music.
Input: D0.b ... 0CH
Output:
Destruction: D0/A0/A2
Remarks: Because PWM sound sources lack a function for stopping sound production on a channel-by-channel

basis, once sound production is started it cannot be stopped.

0DH Requesting to stop an SE

Function: Terminates an SE.
Input: D0.b ... 0DH
Output:
Destruction:
Remarks: Because PWM sound sources lack a function for stopping sound production on a channel-by-channel basis, once sound production is started it cannot be stopped.

Data Request Numbers

For various pieces of data (music and SEs), the following range of request numbers can be specified:

Music	SE
80H~EFH	01H~7FH

The following effect commands can be requested:

Number	Effect name	Effect
F0H	Fade-in	Produces a fade-in effect.
F1H	Fade-out	Produces a fade-out effect.
F2H	Music	Stops the music being played.
F3H	Stopping SEs	Stops all SEs from which sounds are being produced.
F4H	Pausing	Pauses music.
F5H	Resetting a pause	Resets music from a pause state.
F6H	Music master transposing up	Raises the music's master transposition by a halftone.
F7H	Music master transposing down	Lowers the music's master transposition by a halftone.
F8H	SE master transposing up	Raises the SE's master transposition by a halftone.
F9H	SE master transposing down	Lowers the SE's master transposition by a halftone.
FAH	Music master volume up	Increases the master volume for music by 1.
FBH	Music master volume down	Reduces the master volume for music by 1.
FCH	SE master volume up	Increases the master volume for the SE by 1.
FDH	SE master volume down	Reduces the master volume for the SE by 1.
FEH/FFH	Sound Driver initialization	Initialize the Sound Driver.

Sound Data

This section describes the internal structure of sound data. For address specification, the term "address" refers to a relative address from the starting address shown in the following table.

Top Vector

A top vector stores an offset address of data (4 bytes per address).

Address of PCM information
Address of the PWM information table
Address of SE information
Address of the SE information table
Address of table envelope data
Address of table vibrato data
Address of FM rhythm kit data
Address of FM sound source timbre data

Address Tables

An address table is contained in PCM, PWM, music, or in an SE, and in one place stores the addresses that point to those pieces of information (waveform addresses and size information in the case of PCM; tempo information and track addresses in the case of music).

PCM Data

PCM data stores addresses for PCM information. Any two-byte data relating to PCM is stored in terms of little-aidian [RITORUEIDIAN - UNKNOWN TERM]

Playback speed (simply a Z80 weight value, not a sampling rate) [1 byte]

Reserved by the system [1 byte]
Address of the top address [4 bytes]
Data size [2 bytes]

Music Information Data

This data codes the information necessary for playing back a piece of music (tempo, address to sequence data).

NTSC/PAL tempo (in the order of NTSC and PAL) [2 x 2 bytes]
Address of PCM sound source/0CH sequence data [2 bytes]

PCM sound source, 0CH, master transpose [1 byte]
PCM sound source, 0CH, master volume [1 byte]
(PCM sound source, 1CH, by driver selection), [4 bytes]
FM sound sources 0-5 (FM 5CH currently not available) [4x6 bytes]
PSG sound sources 0-2+1 (noise channels) [4x4 bytes]
PWM sound sources 0-3 [4x4 bytes]

SE Information

This field provides the data (number of required channels, etc.) necessary for producing SE sounds.

Number of required channels [1 byte]
Priority [1 byte]
Sound source ID for track 0 (*1) [2 bytes]
Sequence data address for track 0 [2 bytes]
Transpose for track 0 [1 byte]
Volume level for track 0 [1 byte]

(*1) See the section on "Identifying sound sources".

Table Vibrato/Table Envelope

These tables require a minimum size of 100H per table (fixed). Both data and commands are coded in numerical values. Vibrato data is expressed in 2's complements as 7FH (maximum positive number), ~00H (neutral), and ~83H (maximum negative number). Only the values 00H~7FH can be used in an envelope. Commands

are expressed in 80H~84H as indicated below:

Number	Function
80H	Returns to the beginning of the table.
81H	Retains the last value.
82H	Moves to a specified table position.
83H	The vibrato moves the sound-generation mode to a neutral position. The envelope stops the sound-generation operator.

FM Drum Kit

The data sequence number is used as a key number during sequencing.

Offset	Parameter name
00H	Timbre number
01H	Volume
02H	Musical interval

03H	Pan-pot
04H	Table vibrato number
05H~07H	System reserve

FM Sound Source Parameters

FM sound source timbre data is in a partially packed format so that it can be written directly into a register.

Offset	Parameter name
00H	Connection/Feedback
01H	Detune/Multiple (slots 1, 3, 2, 4)
05H	Key Scaling/Attack Rate (slots 1, 3, 2, 4)
09H	AM/Dcay Rate (slots 1, 3, 2, 4)
0DH	Sustain Rate (slots 1, 3, 2, 4)
11H	Sustain Level/Release Rate (slots 1, 3, 2, 4)
15H	Total Level (slots 1, 3, 2, 4)

Details of Sequence Commands

The following sequence commands can be used in music and SEs:

01H-7FH Tone length
Function: Sets the tone length. Given a musical interval, a sound can be produced solely on the basis of its length.
Remarks:

80H: Pause code
Function: Sets the pause code. Suspends reading data for a specified interval of time.
Remarks:

81H-8CH Scale (C,C#,D,D#,E,F,F#,G,G#,A,A#,B)
Function: Specifies a musical interval in an octave range. Given a tone length, a sound can be produced solely on the basis of its musical interval.
Remarks:

C0H, data.b
Function: Writes communication data.
Remarks:

C1H, number.b
Function: Requests an SE.
Remarks:

C2H, offset.w, byte-count, data.b, ...
Function: Writes a specified byte count to the offset for a specified sound work space.
Remarks: The Sound Driver does not provide for malfunction that may occur as a result of using this command to rewrite the work space.

C3H, number.b
Function: Sets the FM sound source envelope to the SSG type.
Remarks: For details, see the "YM-2612 Application Manual".

C4H, PMS/AMS data.b
Function: Sets a PMS/AMS
Remarks:

C5H, bank.b, register number.b, data.b
Function: Directly rewrites the register for an FM sound source.

Remarks: For details, see the "YM-2612 Application Manual". The Sound Driver does not provide for malfunction that may occur as a result of using this command to rewrite the work space.

C6H, mode.b

Function: Changes FM source sound production methods. The 0 mode is the ordinary sound production mode; the non-0 mode is the DRUM mode.

Remarks: For a description of the DRUM mode, see the "Tone Editor Manual".

D0H-DFH

Function: Sets the velocity.

Remarks: Velocities are converted as shown below and added to the sound volume level:

Command number	D0H	D1H	D2H	D3H	D4H	D5H	D6H	D7H
Actual value	3CH	38H	34H	30H	2CH	28H	24H	20H
Command number	D8H	D9H	DAH	DBH	DCH	DDH	DEH	DFH
Actual value	1CH	18H	14H	10H	0CH	08H	04H	00H

E0H, number.b

Function: Changes timbres and envelopes.

Remarks:

E1H, absolute-volume.b

Function: Sets the absolute volume. The higher the numerical value, the smaller the volume.

Remarks: The absolute volume is specified in a 00H~7FH range for all sound sources.

E2H, relative-volume.b

Function: Sets the relative volume. The higher the numerical value, the smaller the volume.

Remarks: The absolute volume is specified in a 00H~7FH range for all sound sources. The function does not check for an overflow.

E3H, point.b

Function: Sets the pan-pot. If the Qsound is used on a PWM sound source, the commands sets a Qsound point in MIDI-standard-based numerical values (00H~40H~7FH).

Remarks: This command is applicable only to FM and PWM sound sources.

E4H, tune.b

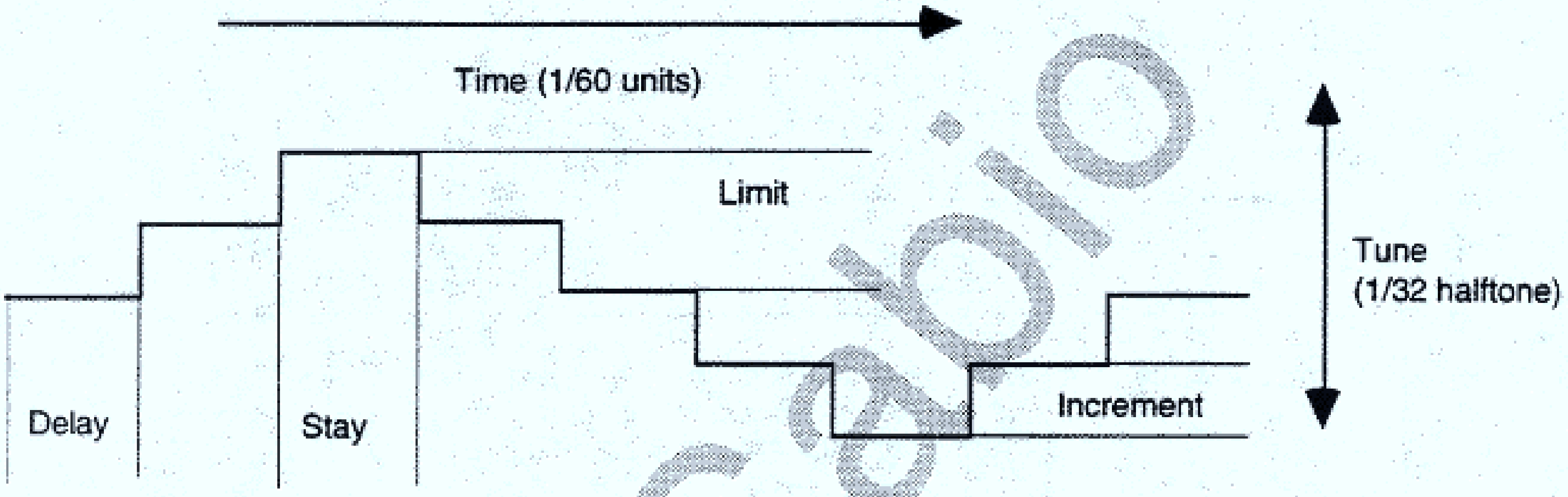
Function: Sets the detune option in units of 1/32 halftone.

Remarks:

E5H, delay.b, stay.b, increment.b, limit.b

Function: After a sound is produced, waits for a time interval equal to a delay interrupt; adds the increment for each stay, and changes the sign when the limit is reached.

Remarks:



E6H, transpose.b

Function: Performs a transpose. Negative values (2's complements) can also be used in this command.

Remarks:

E7H, bend-value.w

Function: Performs a bend. Actually, the bending operation is the 16-bit version of the detuning operation.

Remarks: Because of a 3-bit left shift that is performed internally, the actual resolution is 13 bits.

E8H, number.b

Function: Sets the table vibrato.

Remarks: The 0 value specified in the number field resets the vibrato.

E9H, switch.b

Function: Temporarily rests the vibrato. An 0 resets and a non-zero sets the vibrato.

Remarks: This command is required in order to enable the commands E5H/E8H.

EAH, octave.b
Function: Sets the octave in an absolute value.
Remarks:

EBH, octave.b
Function: Increases the octave by one.
Remarks: Does not check for an overflow.

ECH, octave.b
Function: Decreases the octave by one.
Remarks: Does not check for an overflow.

EEH, mode.b [,data ...]
Function: This command performs different operations, depending on the sound source involved, as follows:

Sound source	Operation	Data byte count
FM sound source (2CH)	Sets the sound effect mode. Specify the immediate value to be written to the register, followed by a Block/F-Number (2 bytes per slot).	5 bytes
FM sound source (5CH)	Allows you to select whether a channel is to be used as a PCM or FM sound source. Specify the immediate value to be written to the register.	1 byte
PSG sound source	Sets a noise. Specify the immediate value to be written to the register.	1 byte
PWM sound source	Switches the Qsound. An 0 turns the Qsound off, a non-0 value turns it on.	1 byte

Remarks: For details, see the respective sound source manuals.

F0H, NTSC.w, PAL.w
Function: Sets the tempo.
Remarks: "NTSC" stores the value determined according to the following formula:
((tempo/150) x $100 + (remainder of tempo/150) x $100/150).w
The value of PAL is NTSC value x 6/5.
This command is not applicable to the SE (the tempo is fixed at 150).

F1H
Function: Disables turning the next note off.

Remarks:

F2H, gate.value.b

Function: Sets the gate.

Remarks:

F3H, address.w

Function: Stores the current address in a work space and moves it to a specified position.

Remarks: An address (including a header) is a relative value from the beginning of music data.

F4H

Function: Moves a given address to a stored address, discarding the stored address.

Remarks: Fetches addresses on a first-in, first-out basis.

F5H-F7H, count.b

Function: Moves to an address, specified by FBH~FDH, a specified number of times.

Remarks: This is a repeat function. There are three commands of this type. Consequently, a maximum of three nesting levels can be used.

F8H-FAH

Function: Terminates the repeat count specified by F5H~F7H, and moves to the data following F5H~F7H.

Remarks:

FBH-FDH

Function: Saves the starting repeat position.

Remarks: This command is meaningful only if used in conjunction with the commands F5H~F7H.

FEH

Function: Saves the current address.

Remarks: This command is meaningful only if 1 is specified in FEH.

FFH, mode.b

Function: This command indicates the end of track data. If a non-0 mode is in effect, control moves to the data following FEH.

Remarks: In the case of a non-0 mode, use FEH to set a move-to point.

*: This data is applicable only to the Driver. Not all data can be controlled by means of data that is based on the MIDI converter.

Identifying Sound Sources

The following IDs are assigned to various sound sources:

Sound source	FM0	FM1	FM2	FM3	FM4	FM5	PSG0	PSG1	PSG2	PSGN
ID	00H	01H	02H	04H	05H	06H	80H	A0H	C0H	E0H
Sound source	PCM0	PCM1	PWM0	PWM1	PWM2	PWM3				
ID	40H	41H	08H	0AH	0CH	0EH				

Notes

The following ranges of data can be used as pseudo-V3:

Item	Range	Remarks
Key	00H~7FH	For note data keys, the octave and the scale are set separately. Keys are defined according to MIDI standards. Keys that cannot be implemented in hardware (e.g., octaves 1 and 9) cannot be played correctly.
Volume	00H~7FH	The maximum allowable volume is 00H, the minimum 7FH. The sound sources are implemented according to their hardware specifications without the balancing of volume levels. In some sound sources, other than the FM sound source, a specified volume level cannot be produced because of hardware/software limitations.
Timbre	00H~7FH	The timbre parameter is not applicable to PCM or PWM. For PSG, a timbre is treated as a software envelope number.
Pan-pot	00H~7FH	The allowable range of numeric values for the pan-pot category is based on MIDI standards. This range is not applicable to sound sources (i.e., PCM and PSG) that do not have a pan-pot due to hardware/software limitations.
Sound Driver address		Any address that is executable by the 68000 is allowed.
Sound work space	FFF000H	This is fixed as a general rule, and cannot be modified.

Maximum number of sounds produced	15~16	This number varies with the particular PCM driver selected.
Number of SE tracks	5	This is fixed as a general rule, and cannot be modified.
Switching between FM5 and D/A	–	Switching between FM and DA can be performed either by not entering any data in the FM5 or by using the command $EE.
YM-2612 write-in data buffer	200H	This is fixed as a general rule, and cannot be modified.